

**Hock, KM (Kai)**

---

**From:** Meng Hock, K (Kai)  
**Sent:** 02 August 2007 17:39  
**To:** Wolski, A (Andy); Panagiotidis, K (Kosmas)  
**Subject:** RE: clapack

The server thinks that gpp\_dgesvd-ex.bat is dangerous and has removed it. Anyway, it just contains the following command line for linking and compiling clapack:

```
g++ -o dgesvd-ex.exe dgesvd-ex.cpp -L. -llapack -lblas -IF77 -lI77
```

Kai

---

**From:** Meng Hock, K (Kai)  
**Sent:** 02 August 2007 17:35  
**To:** Wolski, A (Andy); Panagiotidis, K (Kosmas)  
**Subject:** clapack

Hi Kosmos, Andy,

I have written an example program for SVD using clapack. The files attached are:

data	dgesvd-ex.d
program	dgesvd-ex.cpp
output	dgesvd-ex.r
command line linking:	gpp_dgesvd-ex.bat

It is not very easy to use, but we can always write a wrapper to pass the arguments for the Numerical Recipes function to the clapack function, like I have done for Desmond's spin tracking program.

Another thing is that it is actually quite difficult to install. The procedures given with the source codes are meant for a specific compiler. To use it for our own compiler may require a bit of work.

I am using the MinGWWin C++ compiler (g++). It took a few days, and I actually have to modify some of the source codes, but I finally managed to get it up and running. The steps listed below are what I have learnt the hard way.

Hopefully, it would not be too much trouble to adapt this to Visual C++. I have used the matrix inversion function before and it is much faster than what I got from Numerical Recipes, so I think it is well worth trying.

Best wishes,

Kai.

=====  
CLAPACK: INSTALLING, USING  
=====

DOWNLOAD SOURCES  
=====

1. clapack source code

<http://www.netlib.org/clapack/index.html>

CLAPACK3-Windows.zip

2. MinGWWin C++ compiler (not needed if you use other compilers):

<http://mingw.org/download.shtml>

3. Which clapack function?

[www.mathworks.com/access/helpdesk/help/techdoc/ref/svd.html](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/svd.html)

4. Simple CLAPACK Examples:

<http://theochem.mercer.edu/clapack/>

5. LAPACK (Fortran) examples, documentation

<http://www.nag.co.uk/lapack-ex/node128.html>

<http://www.netlib.org/lapack/lug/node32.html>

6. Writing make files:

<http://www.eng.hawaii.edu/Tutor/Make/>  
<http://users.actcom.co.il/%7Echoo/lupg/tutorials/writing-makefiles/writing-makefiles.html>

## 7. static C library:

<http://users.actcom.co.il/~choo/lupg/tutorials/libraries/unix-c-libraries.html>

## 8. Free unzip software

<http://www.7-zip.org/>

## INSTALLATION

=====

### 1. Unzip CLAPACK3-Windows.zip.

This contains a folder called CLAPACK with a number of subfolders.

### 2. Make four C static libraries from source codes.

These are: liblapack.a  
 libblas.a  
 libF77.a  
 libl77.a

### 3. To make liblapack.a, go to .\CLAPACK\SRC Each file contains one function of clapack.

Locate the file "Makefile" and open with text editor.  
 This contains the names of all the function files in the folder.

However, this make file has been written for a specific C compiler.  
 You must go through it properly and edit it for your own compiler.

If you are using g++ (C++ compiler) from MinGWWin,  
 then you can get all the make files from me.

Note that the definitions used in the Makefile are given in the file "make.inc" located in .\CLAPACK\

Run this make file and create liblapack.a

If you are not familiar with static libraries and make files, have a look at DOWNLOAD SOURCES nos. 6 and 7 above.

4. To make libblas.a, go to .\CLAPACK\BLAS\SRC. Locate the file "Makefile" and open with text editor.

This contains the names of all the function files in the folder. You must go through it properly and edit it for your own compiler.

Run this make file and create libblas.a

5. To make libF77.a, go to .\CLAPACK\F2CLIBS\libF77. Repeat the above steps and create libF77.a

6. To make libl77.a, go to .\CLAPACK\F2CLIBS\libl77. Repeat the above steps and create libl77.a

7. You may encounter problems that are specific to your compiler. For g++, I had the following problems:

- (i) Certain variables used may be operators for some compilers.

E.g. stgevc.c in .\CLAPACK\SRC uses the variable compl. This is the operator for finding complementary of binary variables in g++.

So g++ gives the not altogether obvious error message for line 239: expected primary-expression before "static"

It took me a while to guess that it is complaining about the variable on line 239, which happens to compl  
- I did not know it is an operator.

- (ii) In the codes in .\CLAPACK\F2CLIBS\libl77,

many functions are written without a type.

g++ is not happy with this, so I had to insert all by hand.  
I do it one by one as the make command gives an error message.

## USING CLAPACK =====

1. Create a working folder, say .\test and copy the four C libraries into it.

Go to .\CLAPACK\SRC and locate the files blaswrap.h and f2c.h.  
Copy these into .\test.

2. Suppose your program name is main.cpp, and you wish to call a clapack function from this.

Be sure to insert on top the lines:  
#include blaswrap.h  
#include f2c.h

3. Suppose the clapack function you want to use is dgesvd\_.  
Remember to insert the function prototype before calling the function.

For this purpose, you can locate the source code in dgesvd.c  
in folder .\CLAPACK\SRC

4. To compile and link, if you have g++, use

```
g++ -o main.exe main.cpp -L. -llapack -lblas -IF77 -lI77
```

5. Important note:

(i) Documentation for the function is given in the file, say dgesvd.c

(ii) 2D array in clapack are arranged in 1D array, with columns contiguous.  
The documentation for the array would look like 2D -

that is because it was meant for the Fortran version.

- (iii) An easy way to determine which function you need is to use the help facility for Matlab (DOWNLOAD SOURCES no. 3 above).

6. Again you may encounter problems specific to your compiler.  
For g++, I had the following problems:

- (i) The functions in `.\CLAPACK\SRC` often define a prototype even for standard maths functions like `sqrt`, `log`, etc.

g++ is not happy with this, so I had to remove all by hand.  
The line `#include "math.h"` must also be added to the top of the corresponding file.

I do it one by one as the link command gives an error message.